1. What are the fundamentals of vulnerability management?

Fundamentals of Vulnerability Management

Vulnerability management (VM) is an essential **proactive security process** to identify, assess, prioritize, and mitigate security weaknesses in an organization's systems, applications, and networks.

🔹 1. Identification of Vulnerabilities

The first step in vulnerability management is **discovering security weaknesses** in IT assets.

✅ Methods of Identification:

🔍 **Automated Scanning:**

- Use tools like **Nessus, OpenVAS, Qualys, Burp Suite** to scan for known vulnerabilities.

🔍 **Manual Penetration Testing:**

- Ethical hackers simulate real-world attacks to uncover security flaws.

🔍 **Code Analysis:**

- **Static (SAST)** – Reviewing source code (e.g., SonarQube, Semgrep).
- **Dynamic (DAST)** – Testing running applications (e.g., OWASP ZAP).

🔍 **Asset Inventory:**

- Maintain an updated **list of all software, hardware, and cloud assets** to know where vulnerabilities might exist.

🔍 **Threat Intelligence:**

- Monitor CVE databases (**NVD, Exploit-DB**) and threat feeds for new vulnerabilities.

🔹 2. Assessment & Prioritization

Not all vulnerabilities are equally critical. After identifying weaknesses, security teams must **assess risk and prioritize fixes**.

✅ How to Assess Risk:

🛑 **CVSS Scoring (Common Vulnerability Scoring System):**

- Rates vulnerabilities on a **scale from 0 to 10**, based on severity.
- Example:
  - **9.8+ (Critical)** – Remote code execution (RCE)
  - **7.0 - 8.9 (High)** – Privilege escalation
  - **4.0 - 6.9 (Medium)** – Information leakage
  - **0.1 - 3.9 (Low)** – Minor misconfigurations

🛑 **Business Impact Analysis:**

- Does the vulnerability affect **critical services** (e.g., banking, healthcare)?
- Could exploitation cause **financial loss or reputational damage**?

🛑 **Exploitability & Threat Intelligence:**

- Check **Exploit-DB, Metasploit, and CISA KEV Catalog** for active exploits.
- If a vulnerability is being actively **exploited in the wild**, fix it immediately!

🔹 3. Remediation & Mitigation

Once vulnerabilities are prioritized, teams must take **action to eliminate or reduce risks**.

✅ Remediation Strategies:

✔️ **Apply Security Patches:**

- Always **update software, firmware, and OS** to patch known vulnerabilities.
- Example:

```
sudo apt update && sudo apt upgrade -y
```

✔️ **Configuration Hardening:**

- Disable **unused services, default accounts, and weak encryption**.
- Enforce **least privilege access** and **secure authentication**.

✔️ **Workarounds & Temporary Fixes:**

- If patches aren't available, use **firewall rules, WAFs, or IPS signatures** to block attacks.
- Example:
  - Use a **mod_security rule** to block SQL injection attempts.

✔️ **Zero Trust Security:**

- Assume **no network is safe** and enforce strict authentication controls.

🔹 4. Continuous Monitoring & Reassessment

Vulnerability management is an **ongoing process**. New threats emerge daily, so continuous **monitoring and reassessment** are required.

✅ How to Maintain Security:

🔄 **Regular Vulnerability Scans:**

- Automate scans **weekly or monthly** to detect new risks.

🔄 **Security Patch Management:**

- Establish a **patching schedule** for OS, applications, and cloud services.

🔄 **Security Audits & Compliance Checks:**

- Align with security frameworks like **ISO 27001, NIST, CIS Controls**.

🔄 **Incident Response & Threat Hunting:**

- Monitor logs with **SIEM tools (Splunk, Graylog, Wazuh)**.
- Investigate **suspicious activities** using **Wireshark or Zeek**.

🔹 5. Reporting & Documentation

Proper **documentation** ensures that security teams can track vulnerabilities, measure progress, and improve security posture over time.

✅ What to Include in Reports:

📌 **Discovered vulnerabilities** – Type, affected system, CVSS score.
📌 **Risk assessment** – Exploitability, business impact.
📌 **Mitigation actions** – Patches applied, configurations changed.
📌 **Incident history** – Record previous security breaches and lessons learned.
📌 **Compliance adherence** – Document adherence to security regulations (GDPR, HIPAA, PCI-DSS).

🔹 6. Security Awareness & Training

Human errors **often introduce vulnerabilities** (e.g., weak passwords, phishing attacks). Regular **security training** helps prevent these issues.

✅ Key Training Topics:

👨‍💻 **Phishing Awareness** – Recognizing fake emails & social engineering.
👨‍💻 **Secure Coding Practices** – Writing safe code to prevent injection attacks.
👨‍💻 **Incident Response Procedures** – How to react to security breaches.
👨‍💻 **Regular Red Team vs. Blue Team Drills** – Simulating real-world attacks.

🔹 Summary: The Vulnerability Management Lifecycle

🛠 **1. Identify** – Scan & detect vulnerabilities using automated and manual methods.

🔍 **2. Assess** – Prioritize vulnerabilities based on CVSS, exploitability, and business impact.

🏗 **3. Remediate** – Apply patches, harden configurations, or use temporary fixes.

🔄 **4. Monitor** – Continuously scan, audit, and track security improvements.

📝 **5. Document & Report** – Maintain logs, compliance records, and improvement plans.

🛡 **6. Train & Improve** – Conduct security awareness programs to reduce human-related risks.

🚀 Final Takeaway:

A strong vulnerability management program **prevents breaches, reduces risks, and ensures compliance**. By following these fundamentals, organizations can **stay ahead of attackers and protect their critical assets.**